INF581 – Advanced Machine Learning and Autonomous Agents

Patrick Loiseau (Inria)

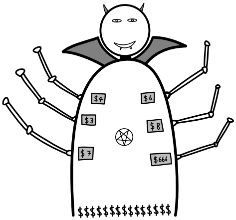
Lecture 3 (part 2/2): Adversarial bandits (and games)

—

Last update: January 18, 2022

Stochastic *versus* adversarial bandits

- Recap: stochastic bandits
  - Arms give reward iid from unknown distribution
  - Typical algorithm UCB (and variants): deterministic
  - Regret bounds in $O(\log n)$ (n: time horizon)
- This lecture: adversarial bandits
  - No stochastic assumption on the rewards
    - no sensitivity to assumptions (robustness)
    - rewards chosen by an adversary
  - In the adversarial setting, we discuss:
    - Algorithms (Exp3)
    - Regret Analysis (in $O(\sqrt{n})$)
    - Extensions
    - Connection to games



[Picture from Lattimore & Szepesvári]

- Note: Markovian bandits (a third kind of bandits)
  - Very different techniques (MDP, DP), not covered here

Outline

Outline

Setup of the k-armed adversarial bandit

- $k > 1$ arms ($=$ space of actions is $[k] := \{1, \cdots, k\}$)
- **Adversary** chooses arbitrary sequence of rewards $(x_t)_{t=1,\cdots,n}$
  (Assume that $x_t \in [0,1]^k$ for all t)
- In each round $t = 1, 2, \cdots, n$
  - **Learner** chooses a distribution $P_t \in \mathcal{P}_{k-1}$ over arms
    Learner samples $A_t \sim P_t$
  - **Learner** observes reward $X_t = x_{tA_t}$

Note: We consider an oblivious adversary, not a *reactive* (or *non-oblivious*) one.

Policy and regret

- Policy: mapping from history sequences to distributions over arms
  Formally $\pi : ([k] \times [0,1])^* \to \mathcal{P}_{k-1}$
  - P_t can depend on actions and rewards up to time $t-1$

- Regret for a given reward sequence $x = (x_t)_{t=1,\cdots,n}$

$$R_n(\pi, x) = \max_{i \in [k]} \sum_{t=1}^{n} x_{ti} - \mathbb{E}\left[\sum_{t=1}^{n} x_{tA_t}\right]$$

  - Note 1: randomization only on the learner's action choice
  - Note 2: this regret makes sense for oblivious adversaries
  - Note 3: sometimes called pseudo-regret

- We want algorithms that do well on worst-case regret:

$$R_n^*(\pi) = \sup_{x \in [0,1]^{n \times k}} R_n(\pi, x)$$

Algorithms from stochastic bandits for adversarial bandits

- Can we use a deterministic policy for adversarial bandits?
  No. For any deterministic policy π, $R_n^*(\pi) \geq n(1 - 1/k)$ (linear)
  - construct a bandit s.t. $x_{tA_t} = 0$ for all t and $x_{ti} = 1$ for $i \neq A_t$
- What about a policy π for adversarial bandits in a stochastic one?
  - Reward X_{ti} drawn from distribution ν_i iid at each t

$$R_n^*(\pi) \geq R_n(\pi, \nu) = \underbrace{\max_{i \in [k]} \mathbb{E}\left[\sum_{t=1}^{n} (X_{ti} - X_{tA_t})\right]}_{\text{stochastic regret}}$$

  - From lower-bounds for stochastic bandits we get that

$$\inf_{\pi} R_n^*(\pi) \geq O(\sqrt{nk})$$

Outline

Key ingredient: importance-weighted estimators

- *Bandit feedback*: observe reward only for chosen arm, $X_t = x_{tA_t}$
- How to estimate the reward for other arms?

- Reminder/notation: P_t is the distribution at t conditioned on history up to $t-1$
  - $P_{ti} = \mathbb{P}(A_t = i | A_1, X_1, \cdots, A_{t-1}, X_{t-1})$
  - We denote $\mathbb{E}_{t-1} Z = \mathbb{E}(Z | A_1, X_1, \cdots, A_{t-1}, X_{t-1})$

- Importance weighted estimator: for all t and $i \in [k]$

$$\hat{X}_{ti} = \mathbb{1}_{A_t = i} \cdot \frac{X_t}{P_{ti}}$$

  - It is an unbiased estimate of x_{ti}: $\mathbb{E}_{t-1} \hat{X}_{ti} = x_{ti}$
  - It has variance $\mathbb{V}_{t-1}[\hat{X}_{ti}] = x_{ti}^2 \cdot \frac{1-P_{ti}}{P_{ti}}$

Another importance-weighted estimator (the loss view)

- $\mathbb{V}_{t-1}[\hat{X}_{ti}] = x_{ti}^2 \cdot \frac{1-P_{ti}}{P_{ti}}$ explodes if P_{ti} small and x_{ti} not small
- but there are many other unbiased estimators...

- The loss view (equivalent to the reward view):
  - Define $y_{ti} = 1 - x_{ti}$, $Y_t = 1 - X_t$
  - Importance-weighted estimator: $\hat{Y}_{ti} = \mathbb{1}_{A_t=i} \cdot \frac{Y_t}{P_{ti}}$
    - ★ unbiased estimator of y_{ti}
    - ★ variance $\mathbb{V}_{t-1}[\hat{Y}_{ti}] = y_{ti}^2 \cdot \frac{1-P_{ti}}{P_{ti}}$

- Immediately gives another estimator for x_{ti}: $1 - \hat{Y}_{ti} = 1 - \mathbb{1}_{A_t=i} \cdot \frac{1-X_t}{P_{ti}}$

Estimator	Variance	Range
$\hat{X}_{ti} = \mathbb{1}_{A_t=i} \cdot \frac{X_t}{P_{ti}}$	$x_{ti}^2 \cdot \frac{1-P_{ti}}{P_{ti}}$	$[0, \infty)$
$\hat{X}_{ti} = 1 - \mathbb{1}_{A_t=i} \cdot \frac{1-X_t}{P_{ti}}$	$(1 - x_{ti})^2 \cdot \frac{1-P_{ti}}{P_{ti}}$	$(-\infty, 1]$

The Exp3 algorithm: Main elements

- Initialize P_1, then for each t:
  - Use an importance-weighted estimator $\hat{X}_{si}$ to estimate the reward for each arm
  - Compute the sum $\hat{S}_{ti} = \sum_{s=1}^{t} \hat{X}_{si}$ (also denoted by $\hat{S}_{t,i}$)
  - Map into a probability distribution that assigns higher weight to more rewarding arms, e.g., by exponential weighting

$$P_{ti} = \frac{\exp(\eta \hat{S}_{t-1,i})}{\sum_{j=1}^{k} \exp(\eta \hat{S}_{t-1,j})}$$

- $\eta > 0$: learning rate
  - η large: close to a max function (exploits aggressively)
  - η close to zero: close to uniform (explores more)
- Here we allow η to depend on k and n (i.e., horizon known in advance)
  - Can be relaxed: doubling trick, decreasing learning rate

The Exp3 algorithm

Input: n, k, η
Set $\hat{S}_{0i} = 0$ for all i
for $t = 1, \ldots, n$ **do**
    Calculate the sampling distribution P_t:

$$P_{ti} = \frac{\exp\left(\eta \hat{S}_{t-1,i}\right)}{\sum_{j=1}^{k} \exp\left(\eta \hat{S}_{t-1,j}\right)}$$

    Sample $A_t \sim P_t$ and observe reward X_t
    Calculate $\hat{S}_{ti}$:

$$\hat{S}_{ti} = \hat{S}_{t-1,i} + 1 - \frac{\mathbb{I}\left\{A_t = i\right\}(1 - X_t)}{P_{ti}}$$

end for

[From Lattimore & Szepesvári]

A first regret bound

Theorem

Let π be the policy of Exp3 with learning rate $\eta = \sqrt{\log(k)/(nk)}$. Then for any $x \in [0,1]^{n \times k}$ we have

$$R_n(\pi, x) \leq 2\sqrt{nk\log(k)}.$$

Remarks:

- The learning rate depends on the time horizon
- Regret bound in $O(\sqrt{nk\log(k)})$: factor $\log(k)$ from the lower bound
    - Can be removed with more sophisticated algorithms[1]

[1] See, e.g., [Lattimore & Szepesvári, p. 157, Note 5].

Proof (1/2)

Let $R_{ni} = \sum_{t=1}^n x_{ti} - \mathbb{E}\sum_{t=1}^n x_{tA_t}$. We will bound R_{ni} for all i. Let $i \in [k]$.

1. By rearranging + tower rule, we have

$$R_{ni} = \mathbb{E}\left[\hat{S}_{ni} - \hat{S}_n\right], \text{ where } \hat{S}_{ni} = \sum_{t=1}^n \hat{X}_{ti} \text{ and } \hat{S}_n = \sum_{t=1}^n \sum_{i=1}^k P_{ti}\hat{X}_{ti}.$$

2. By the telescoping argument, we show a bound on $\exp(\eta \hat{S}_{ni})$:

$$\exp(\eta \hat{S}_{ni}) \leq k \prod_{t=1}^n \frac{W_t}{W_{t-1}}, \text{ where } W_t = \sum_{j=1}^k \exp(\eta \hat{S}_{tj}).$$

3. By exploiting the inequalities

$$\exp(x) \leq 1 + x + x^2 \text{ for all } x \leq 1 \text{ and } 1 + x \leq \exp(x) \text{ for all } x \in \mathbb{R}$$

show that

$$\frac{W_t}{W_{t-1}} \leq \exp\left(\eta \sum_{j=1}^k P_{tj}\hat{X}_{tj} + \eta^2 \sum_{j=1}^k P_{tj}\hat{X}_{tj}^2\right)$$

Proof (2/2)

Recall: we want to upper bound $R_{ni} = \mathbb{E}\left[\hat{S}_{ni} - \hat{S}_n\right]$ for an arbitrary $i \in [k]$

4. By combining 2 and 3 above, taking the log and dividing by η, we get

$$\hat{S}_{ni} - \hat{S}_n \leq \frac{\log(k)}{\eta} + \eta \sum_{t=1}^{n} \sum_{j=1}^{k} P_{tj} \hat{X}_{tj}^2$$

5. By a computation similar to the variance computation, we get

$$\mathbb{E} \sum_{j=1}^{k} P_{tj} \hat{X}_{tj}^2 \leq k$$

6. Summing over t, we get

$$R_{ni} \leq \frac{\log(k)}{\eta} + \eta n k$$

7. Optimizing over η leads to $\eta = \sqrt{\log(k)/(nk)}$ and to the result

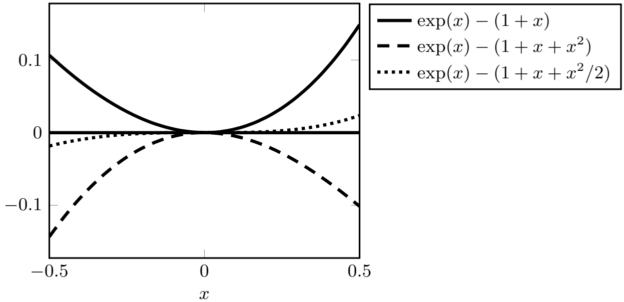
A slight improvement of the regret bound

Theorem

Let π be the policy of Exp3 with learning rate $\eta = \sqrt{2\log(k)/(nk)}$ (instead of $\sqrt{\log(k)/(nk)}$). Then for any $x \in [0,1]^{n\times k}$ we have

$$R_n(\pi, x) \leq \sqrt{2nk\log(k)} \ \textit{(instead of } 2\sqrt{nk\log(k)}\textit{)}.$$

- Using a different approximation of $\exp(x)$



[Picture from Lattimore & Szepesvári]

Anytime bound with a decreasing learning rate

Theorem

Let π be the policy of Exp3 with learning rate $\eta_t = \sqrt{\log(k)/(tk)}$. Then for any $x \in [0,1]^{n\times k}$ we have

$$R_n(\pi, x) \leq \sqrt{2nk\log(k)}.$$

- This is called an anytime bound (valid for any n, does not need to know the time horizon)

Proof:

- With a similar proof as before, we show that

$$R_n(\pi, x) \leq \frac{\log(k)}{\eta_n} + \frac{k}{2}\sum_{t=1}^{n} \eta_t$$

- Conclude noting that $\sum_{t=1}^{n} 1/\sqrt{t} \leq \int_0^n 1/\sqrt{t}\mathrm{d}t = 2\sqrt{n}$

The full information case

- Full information setting: at each t observe x_{ti} for all $i \in [k]$
  - Not just the arm chosen
  - Often called prediction with expert feedback

Theorem

Let π be the policy of Exp3 using the actual rewards instead of the estimated ones, with learning rate $\eta = \sqrt{2\log(k)/n}$. Then for any $x \in [0,1]^{n \times k}$ we have

$$R_n(\pi, x) \leq \sqrt{2n\log(k)}.$$

Important remarks:

- Often called Hedge algorithm (more generally multiplicative weights)
- We get a logarithmic dependence on k only
- Proof: same but using Hoeffding's lemma instead of the polynomial upper bound on $\exp(x)$

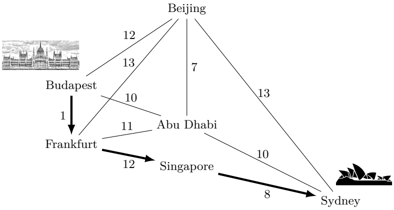
Outline

Combinatorial bandits

- Action set $\mathcal{A} \subset \{0,1\}^d$
  - k exponentially large
  - considering each action as an arm and applying Exp3 is hopeless (for bandit feedback)
- Linear payoff structure: adversary chooses $y_t \in \mathbb{R}^d$

$$R_n = \max_{a \in \mathcal{A}} \mathbb{E}\left[\sum_{t=1}^{n} \langle A_t - a, y_t \rangle\right]$$

- Example: shortest path
- Different feedback
  - full information
  - semi-bandit
  - bandit



[Picture from Lattimore & Szepesvári]

Algorithms for combinatorial bandits

- Bandit feedback: variant of Exp3 with an exploration distribution
- Regret in $O(m\sqrt{nd\log(|\mathcal{A}|)})$, where m is a bound on $|\langle A_t, y_t\rangle|$
- Computational issues
  - Finding a good exploration distribution
  - Sampling from the computed distribution
  - Solutions available in some special cases (e.g., online shortest path)
- Semi-bandit feedback: different algorithms (Exp3, OSMD, FPL)
  - OSMD: regret in $O(\sqrt{nmd(1+\log(d/m))})$
  - Computational issues here too

Some other kinds of bandits

- Linear bandits: $\mathcal{A} \subset \{0, 1\}^d$
- Contextual bandits: at each time step, there is a "context"
  - Typical example: ad placement
- Side observation
- Delayed feedback
- ... and many more

- There exists also other kinds of algorithms (follow the perturbed leader, mirror descent, etc.)
- Connection with online optimization

Outline

Game definition

- A game (in normal form) is a tuple
  - A set of players: A and B (2-player games)
  - A set of actions for each player: $\mathcal{A}$, $\mathcal{B}$ (assume finite)
  - A payoff for each player $i \in \{A, B\}$: $U_i(a, b)$ for any $(a, b) \in (\mathcal{A}, \mathcal{B})$
    - The payoff of a player depends (also) on the other's action
- Models a wide range of multi-agent "competitive" situations
  - Economics (e.g., auctions), CS (spectrum allocation), security, etc.
- Example: Matching pennies
  - Two players $\{A, B\}$
  - $\mathcal{A} = \mathcal{B} = \{heads, tails\}$
  - Payoffs given by

		Player B	
		heads	tails
Player A	heads	(+1, -1)	(-1, +1)
	tails	(-1, +1)	(+1, -1)

Equilibrium and minmax theorem

- Mixed strategy: distribution over actions: $\sigma_A \in \Delta(\mathcal{A}), \sigma_B \in \Delta(\mathcal{B})$
- Nash equilibrium: every player is at best response
  - Strategy profile (σ_A^*, σ_B^*) such that

$$\sigma_A^* \in \underset{\sigma_A \in \Delta(\mathcal{A})}{\arg\max}\, U_A(\sigma_A, \sigma_B^*) \quad \text{and} \quad \sigma_B^* \in \underset{\sigma_B \in \Delta(\mathcal{B})}{\arg\max}\, U_B(\sigma_A^*, \sigma_B)$$

  - A fixed-point such that no player wants to unilaterally deviate from its choice

- Special case of zero-sum games
  - The sum of payoffs is constant equal to zero

$$U_A(a, b) = -U_B(a, b) \text{ for all } (a, b) \in \mathcal{A} \times \mathcal{B}$$

  - Defined by a single utility $U(a, b) = U_A(a, b) = -U_B(a, b)$
  - Fundamental minimax theorem:

$$\max_{\sigma_A \in \Delta(\mathcal{A})} \min_{\sigma_B \in \Delta(\mathcal{B})} U(a, b) = \min_{\sigma_B \in \Delta(\mathcal{B})} \max_{\sigma_A \in \Delta(\mathcal{A})} U(a, b) \quad [= \text{game value}]$$

  - The minimax strategies form a Nash equilibrium

Link with bandits/regrets (for zero-sum games)

Consider a repeated zero-sum game and assume that Player A is an algorithm playing a no-regret strategy (e.g., Hedge in full information), that is such that $R_n/n \to 0$. Then we can look at two cases:

1. The adversary (Player B) is playing best-response
   - We can show that

$$\max_{\sigma_A \in \Delta(\mathcal{A})} \min_{\sigma_B \in \Delta(\mathcal{B})} U(a,b) \geq \min_{\sigma_B \in \Delta(\mathcal{B})} \max_{\sigma_A \in \Delta(\mathcal{A})} U(a,b) - R_n/n$$

   $\Rightarrow$ gives a proof of the minimax theorem
2. The adversary (Player B) is playing a no-regret strategy
   - The average utilities converge to the game value
   - The average strategies are approximate minimax

For nonzero-sum games, the situation is more complex... See this and (much) more in [Slivkins] Chapter 9

Main general references (with references inside to the original papers)

Books:

- "Bandit Algorithms" [Lattimore & Szepesvári]
  - This lecture is mainly based on Chapter 11
- "Prediction, learning and games" [Cesa-Bianchi & Lugosi]

Surveys:

- "Introduction to Multi-Armed Bandits" [Slivkins]
  - In particular Chapter 9 on the connection to games
- "Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems" [Bubeck & Cesa-Bianchi]